2-1

PL Semantics := a function (relation) that maps programs to meanings

B language $P = T \mid F \mid P \otimes P$

$M = T \mid F$

interpreter := define the fun B come other language, X

axiomatic semantics := define the meaning relation w/ math

premise
conclusion

$$\frac{}{M(T) = T} \qquad \frac{}{M(F) = F} \qquad \frac{M(P_1) = T}{M(P_1 \otimes P_2) = T}$$

$$\frac{M(P_2) = T}{M(P_1 \otimes P_2) = T}$$

denotational semantics := in math, compile P to math

→ operational semantics := interpreter written in math w/ constraints

the type is a relation between programs $(P \times P)$

"reduces" complex Ps to simple ones

$r \subset P \times P$ $\quad (T \otimes P_2) \; r \; T$ $\quad r \quad n \quad x$

$(F \otimes P_2) \; r \; P_2$ $\quad \rightarrow_r \quad \rightarrow_n \quad \rightarrow_x$

reflexive closure of r $\quad (T, T) \in \text{refl}(r)$

$\forall x. \; (x,x) \in R$ $\quad (T \otimes F, T \otimes F) \in \text{refl}(r)$

compatible closure

compat(r)

$$\frac{P_1 \; \text{compat}(r) \; P_2}{P_1 \otimes P_3 \; c(r) \; P_2 \otimes P_3} \qquad \frac{P_1 \; c(r) \; P_2}{P_3 \otimes P_1 \; c(r) \; P_3 \otimes P_2}$$

$(1+1)+(1+1)$

$\downarrow c(r)$

$2 + (1+1)$ $\quad (1+1) + 2$

$$\frac{P_1 \; r \; P_2}{P_1 \; c(r) \; P_2}$$

2-2 r = orig (rules)

$\rightarrow_r$ = compatible closure (apply anywhere in program)

$\twoheadrightarrow_r$ = reflexive + transitive -closure (do many times)

$$\frac{P_1 \rightarrow_r P_2}{P_1 \twoheadrightarrow_r P_2} \qquad \frac{}{P_1 \twoheadrightarrow_r P_1} \qquad \frac{P_1 \twoheadrightarrow_r P_2 \quad P_2 \twoheadrightarrow_r P_3}{P_1 \twoheadrightarrow_r P_3}$$

$(F \otimes (F \otimes T)) \twoheadrightarrow_r T$

$=_r$ = symmetric closure

$$\frac{P_1 \twoheadrightarrow_r P_2}{P_1 =_r P_2} \qquad \frac{P_2 =_r P_1}{P_1 =_r P_2}$$

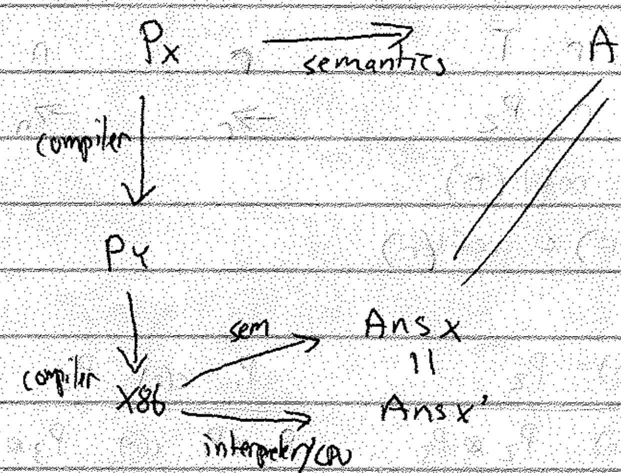
eval$_r$:= a function from P to A (answers)

$\text{eval}_B(P) = T$ iff $P =_B T$ $\quad (T \otimes P_2)\ B\ T$

F iff $P =_B F$ $\quad (F \otimes P_2)\ B\ P_2$

What are semantics for? — $\underbrace{\text{reasoning}}_{\text{asking questions}}$ about programs

asking questions $\longrightarrow$ predictions



2-3

Semantics define program meaning
which facilitates verification of programs
+ program tools (like compilers)

Desirable: Programs are deterministic.

$$\underbrace{\forall P.}_{\text{for all programs in language}} \ \forall A_1 . \forall A_2 , \ \ evalr(P) = A_1 \wedge evalr(P) = A_2 \Rightarrow A_1 = A_2$$

specialized for one P

main:
 x = 10
 thread-create(f, &x)
 thread-create(g, &x)
 wait_threads
 ret x ⇒ 30, 25, 15, 20

locking access to x

```
f(int* x) {    ← lock →    g(int *x) {
  *x = *x + 5;               *x = *x * 2;
}              ← unlock →  }
```

compatible doesn't "where" work happens

$$\begin{tikzcd}
(1+2) + (3+4) \arrow[d] \arrow[dr] & \\
3 + (3+4) \arrow[d] & (1+2) + 7 \arrow[dl] \\
3 + 7 \arrow[r, two heads] & 10
\end{tikzcd}$$

Church-Rosser

$\forall M, N.$
If $M =_r N$, $\exists L.$
$M \twoheadrightarrow_r L$
and $N \twoheadrightarrow_r L$.

Diamond

If $L \twoheadrightarrow M'$ and $L \twoheadrightarrow N'$
then $\exists L'.$ $M' \twoheadrightarrow L'$
and $N' \twoheadrightarrow L'$

$$\begin{tikzcd}
 & L \arrow[dl] \arrow[dr, two heads] & \\
M \arrow[dr, two heads] & & N \arrow[dl] \\
 & L' &
\end{tikzcd} \iff$$